What is claimed is:

1. A method of performing an arbitrary permutation of a source sequence of bits in a programmable processor comprising the steps of:

a. defining an intermediate sequence of bits that said source sequence of bits is transformed into;

b. determining a permutation instruction for transforming said source sequence of bits into said intermediate sequence of bits; and

c. repeating steps a. and b. using said determined intermediate sequence of bits from step b. as said source sequence of bits in step a. until a desired sequence of bits is obtained,

wherein the determined permutation instructions form a permutation instruction sequence.

09850238.050701

2. The method of claim 1 wherein said intermediate sequence of bits is determined with a configuration of a multistage interconnection network.

3. The method of claim 1 wherein said intermediate sequence of bits is determined with a configuration of an omega-flip network, said omega-flip network comprising at least two stages wherein each said stage is either an omega network stage or a flip network stage.

4. The method of claim 3 wherein the permutation instruction comprises an opcode indicating the configuration of said omega-flip network, a reference to a source register which contains said source sequence of bits, a reference to one or more configuration registers which contain configuration bits, and optionally a reference to a destination register to which said intermediate sequence of bits or said desired sequence of bits is placed.

5. The method of claim 4 where said opcode comprises a plurality of bits, with a first bit for indicating if a first said stage is said omega network or said flip network stage and a second bit for indicating if a second said stage is said omega network or said flip network stage.

6. The method of claim 4 wherein said opcode comprises a plurality of bits with each said bit indicating if said stage is said omega network or said flip network.

7. The method of claim 3 wherein the permutation instruction comprises an opcode indicating the configuration of said omega-flip network, a reference to a source register which contains said source sequence of bits, and a reference to oneor more configuration registers which contain configuration bits, and optionally a reference to a destination register to which said intermediate sequence of bits or said desired sequence of bits is placed wherein each said configuration of said omega-flip network has at least two said stages, wherein each said stage uses configuration bits from one of said configuration registers

8. The method of claim 7 wherein a first set of said configuration bits determines movement of said source sequence of bits in said source register to said intermediate sequence of bits and a second set of said configuration bits determines movement of said intermediate sequence of bits into said destination register.

9. The method of claim 7 wherein each of said configuration bits is applied to a pair of conflict outputs of said omega-flip network.

10. The method according to claim 7 wherein said permutation instruction comprises a reference to two configuration registers, and said omega-flip network comprises four stages.

11. The method of claim 1 wherein said intermediate sequence of bits is determined with a configuration of an omega-flip network, said omega flip network comprising a first omega stage connected to a first flip stage, said first flip stage connected to a second flip stage and a said second flip stage connected to a second omega stage wherein said permutation instruction uses two stages selected from the group consisting of said first omega stage, said first flip stage, said second omega stage, said second flip stage and two unselected stages are configured as pass through stages.

09850238 .050701

12. The method of claim 1 wherein said intermediate sequence of bits is determined with a configuration of a modified omega-flip network, said modified omega flip network comprising a first modified omega stage connected to a first modified flip stage, said first modified flip stage connected to a second modified flip stage and a said second modified flip stage connected to a second modified omega stage wherein said modified omega stage is an omega stage with pass throughs and said modified flip stage is a flip stage with pass throughs.

13. The method of claim 3 wherein said configuration of an omega-flip network is determined by the steps of:

determining a configuration of a Benes network for said permutation; and

translating said configuration of a Benes network into said configuration of an omega-flip network.

09850238 .050701

14. The method of claim 13 wherein said permutation instruction is assigned to two or more stages of said omega-flip network.

15. The method of claim 3 wherein at most $2\lg r/m$ said permutation instructions are included in said permutation instruction sequence, wherein *r* is the number of subwords in said sequence of bits, each said subword comprising one or more bits, and m is the number of network stages performed by one permutation instruction.

16. The method of claim 1 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, or programmable System-on-a-Chip (SOC).

17. The method of claim 3 wherein said source sequence of bits is formed in subwords of two or more of said bits and further comprising the steps of:

after step c, determining pass through stages in said omega-flip network; and

eliminating said pass through stages.

18. The method of claim 3 wherein said permutation is reversed by the steps of:

reversing an order of said permutation instructions in said permutation instruction sequence;

reversing an order of each of said omega network stages and said flip network stages in said omega-flip network in each said permutation instruction; and

changing each of said omega network stages to said flip network stages and each of said flip network stages to said omega network stages.

19. The method of claim 1 wherein configuration bits are used in said permutation instruction for determining movement of said source sequence of bits in a source register to said intermediate sequence of bits or movement of said intermediate sequence of bits into a destination register or a second intermediate sequence of bits and further comprising the steps of storing said configuration bits and retrieving said stored configuration bits during step b.

20. The method of claim 1 wherein configuration bits are used in said permutation instruction for determining movement of said source sequence of bits in said source register to said intermediate sequence of bits or movement of said intermediate sequence of bits into said destination register or said source register and further comprising the steps of storing a portion of said configuration bits and retrieving said stored portion of configuration bits during step b.

21. A method of performing an arbitrary permutation of a source sequence of bits in a programmable processor said source sequence of bits is packed into a plurality of source registers comprising the steps of:

a. dividing bits of a first of said source registers to be placed in a first destination register into a first group and bits of said first of said source registers to be placed in a second destination register into a second group with a first permutation instruction sequence;

b. dividing bits of a second of said source registers to be placed in said first destination register into a first group and bits of said second of said source registers to be placed in a second destination register into a second group with a second permutation instruction sequence;

c. placing bits of said first group of said first of said source registers and said bits of said first group of said second of said source registers into said first destination register;

d. placing bits of said second group of said first of said source registers and said second group of said second of said registers into said second destination register;

e. defining a sequence of bits of said first destination register as a first source sequence of bits and a sequence of bits of said second destination register as a second source sequence of bits;

f. defining an intermediate sequence of bits that each of said first source sequence of bits and said second source sequence of bits is transformed into;

g. determining a permutation instruction for transforming said first source sequence of bits and said second source sequence of bits into respective said intermediate sequence of bits; and

repeating steps f. and g. using said determined intermediate sequence of bits from step g. as said source sequence of bits in step f. until a respective desired sequence of bits is obtained for said first source sequence of bits and said second source sequence of bits,

wherein the determined permutation instructions form a permutation instruction sequence.

09850238.050201

22. The method of claim 21 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, or programmable System-on-a-Chip (SOC).

23. The method of claim 18 wherein at most 4lg*n*+2 instructions are included in said permutation instruction sequence, wherein *n* is the number of subwords in said sequence of bits, each said subword comprising one or more bits.

24. A method of performing an arbitrary permutation of a source sequence of bits in a programmable processor comprising the steps of:

a. defining an intermediate sequence of bits that said source sequence of bits is transformed into with a configuration of an omega-flip network, said omega-flip network comprising at least two stages, each said stage is either an omega network stage or a flip network stage;

b. determining a permutation instruction for transforming said source sequence of bits into said intermediate sequence of bits;

c. storing said determined intermediate sequence of bits; and

d. determining a subsequent permutation instruction using said stored intermediate sequence of bits.

25. The method of claim 24 further comprising repeating step c. and step d. until a desired sequence of bits is obtained,

wherein the determined permutation instructions form a permutation instruction sequence.

09850238 .050701

26. The method of claim 25 wherein configuration bits are used in said permutation instruction for determining movement of said source sequence of bits in said source register to said intermediate sequence of bits or movement of said intermediate sequence of bits into a destination register or a second intermediate sequence of bits and two configuration registers are used for storing said configuration bits.

27. A method for performing a permutation of a source sequence of bits in a programmable processor comprising the steps of:

defining an intermediate sequence of bits that said source sequence of bits is transferred into using an interconnection network; and

determining a sequence of one or more permutation instructions for transferring said source sequence of bits into said intermediate sequence of bits and optionally one or more subsequent intermediate sequences of bits until a desired sequence of bits is obtained.

28. A method for performing a permutation of a source sequence of bits in a programmable processor comprising the steps of:

defining an intermediate sequence of bits that said source sequence of bits is transferred into using an omega-flip network; and

determining a sequence of one or more permutation instructions for transferring said source sequence of bits into said intermediate sequence of bits and optionally one or more subsequent intermediate sequences of bits until a desired sequence of bits is obtained.

29. The method of claim 28 wherein said intermediate sequence of bits is determined with a configuration of an omega-flip network, said omega-flip network comprising at least two stages wherein each said stage is either an omega network stage or a flip network stage.

30. The method of claim 29 wherein at most $2\lg r/m$ said permutation instructions are included in said permutation instruction sequence, wherein r is the number of subwords in said sequence of bits, each said subword comprising one or more bits, and m is the number of network stages performed by one permutation instruction.

1047050.83205860

31. A system of performing an arbitrary permutation of a source sequence of bits in a programmable processor comprising:

means for defining an intermediate sequence of bits that said source sequence of bits is transformed into using omega network stages and flip network stages; and

means for determining a permutation instruction for transforming said source sequence of bits into one or more intermediate sequence of bits until a desired sequence of bits is obtained,

wherein each intermediate sequence of bits is used as input to the subsequent permutation instruction and the determined permutation instructions form a permutation instruction sequence.

32. The system of claim 31 wherein the omega-flip network at least two stages wherein each stage is either said omega network stage or said flip network stage.

33. The system of claim 32 wherein the permutation instruction comprises an opcode indicating which two stages of said omega-flip network are used in said permutation

instruction, a reference to a source register which contains said source sequence of bits, a reference to at least one configuration register which contains configuration bits.

34. The system of claim 33 wherein said permutation instruction further comprises a reference to a destination register to which said intermediate sequence of bits or said desired sequence of bits is placed.

35. The system of claim 33 wherein said opcode comprises a first bit for indicating if said omega network stage or said flip network stage is used in a first stage of said omega-flip network and a second bit for indicating if said omega network stage or said flip network stage is used in a second stage of said omega-flip network.

36. The system of claim 33 wherein a first set of said configuration bits determines movement of said source sequence of bits in said source register to said intermediate sequence of bits and a second set of configuration bits determines movement of said intermediate sequence of bits into said destination register or a second intermediate sequence of bits.

09850238 .050701

37. The system of claim 33 wherein said opcode comprises a plurality of bits with each said bit indicating if said stage is said omega network or said flip network.

38. The system of claim 33 wherein each of said configuration bits is applied to a pair of conflict outputs of said omega-flip network.

39. The system according to claim 33 wherein said permutation instruction comprises a reference to two configuration registers, and said omega-flip network comprises four stages.

40. The system of claim 31 wherein said omega flip network comprises a first said omega network stage connected to first said flip network stage, said first flip network stage connected to a second flip network stage and a said second flip network stage connected to a

second omega network stage wherein said permutation instruction uses two stages selected from the group consisting of said first omega network stage, said first flip network stage, said second omega network stage, said second flip network stage and two unselected stages are configured as pass through stages.

41. The system of claim 31 wherein said intermediate sequence of bits is determined with a configuration of a modified said omega-flip network, said modified omega flip network comprising a first modified omega stage connected to a first modified flip stage, said first modified flip stage connected to a second modified flip stage and a said modified second flip stage connected to a second modified omega stage wherein said modified omega stage is said omega stage with pass throughs and said modified flip stage is said flip stage with pass throughs.

104050.083205860

42. The system of claim 31 wherein a configuration of an omega-flip network is determined by:

means for determining configuration of a Benes network for said permutation; and

means for translating said configuration of a Benes network into said configuration of an omega-flip network.

43. The system of claim 42 wherein said permutation instruction is assigned to a pair of stages of said omega-flip network.

44. The system of claim 31 wherein at most $2\lg r/m$ said permutation instructions are included in said permutation instruction sequence, wherein r is the number of subwords in said sequence of subwords in said sequence of bits, each said subword comprising one or more bits, and m is the number of network stages performed by one permutation instruction.

45. The system of claim 31 wherein said programmable processor is a microprocessor, digital signal processor, media processor, multimedia processor, cryptographic processor, or programmable System-on-a-Chip (SOC).

46. The system of claim 31 wherein said source sequence of bits is formed in subwords of two or more of said bits and further comprising:

means for determining pass through stages in said omega-flip network; and

means for eliminating said pass through stages.

47. The system of claim 31 further comprising:

means for reversing an order of said permutation instructions in said permutation instruction sequence;

means for reversing an order of each of said stages in said omega-flip network in each said permutation instruction; and

means for changing each of said omega stages to said flip stages and each of said flip stages to said omega stages

wherein said permutation is reversed.

48. The system of claim 31 wherein configuration bits are used in said permutation instruction for determining movement of said source sequence of bits in said source register to said intermediate sequence of bits or movement of said intermediate sequence of bits into a destination register or a source register and further comprising means for storing said configuration bits and means for retrieving said stored configuration bits for use in said permutation instruction.

49. The system of claim 31 wherein configuration bits are used in said permutation instruction for determining movement of said source sequence of bits in said source register to said intermediate sequence of bits or movement of said intermediate sequence of bits into said destination register or said source register and further comprising means for storing a portion of said configuration bits and retrieving said stored portion of configuration bits for use in said permutation instruction.

09852036.052701

50. A system of performing an arbitrary permutation of a source sequence of bits in a programmable processor said source sequence of bits is packed into a plurality of source registers comprising:

means for dividing bits of a first of said source registers to be placed in a first destination register into a first group and bits of said first of said source registers to be placed in a second destination register into a second group with one said permutation instruction sequence;

means for dividing bits of a second of said source registers going to said first destination register into a first group and bits of a second of said source registers going to a second destination register into a second group with one said permutation instruction sequence;

means for placing bits of said first group of said first of said source registers and said bits of said first group of said second of said source registers into said first destination register;

means for placing bits of said second group of said first of said source registers and said second group of said second of said source registers into said second destination register;

means for defining a sequence of bits of said first destination register, as a first source sequence of bits and a sequence of bits of said second destination register as a second source sequence of bits;

means for defining an intermediate sequence of bits that each of said first source sequence of bits and said second source sequence of bits is transformed into; and

means for determining a permutation instruction for transforming said first source sequence of bits and said second source sequence of bits into one or more respective said intermediate sequence of bits until a respective desired sequence of bits is obtained for said first source sequence of bits and said second source sequence of bits,

wherein each intermediate sequence of bits is used as input to the subsequent permutation instruction and the determined permutation instructions form a permutation instruction sequence.

09850238.050701

51. The system of claim 50 wherein at most $4\lg n+2$ instructions are included in said permutation instruction sequence, wherein n is the number of subwords in said sequence of bits, each said subword comprising one or more bits.

52. A system for performing an arbitrary permutation of a source sequence of bits in a programmable processor comprising:

a. means for defining an intermediate sequence of bits that said source sequence of bits is transformed into with a configuration of an omega-flip network, said omega-flip network comprising at least two stages, each said stage is either an omega network stage or a flip network stage;

b. means for determining a permutation instruction for transforming said source sequence of bits into said intermediate sequence of bits;

c. means for storing said determined intermediate sequence of bits; and

d. means for determining a subsequent permutation instruction using said stored intermediate sequence of bits.

09850238.050701

53. The system of claim 52 further comprising means for repeating c. and d. until a desired sequence of bits is obtained,

wherein the determined permutation instructions form a permutation instruction sequence.

54. The system of claim 52 wherein configuration bits are used in said permutation instruction for determining movement of said source sequence of bits in said source register to said intermediate sequence of bits or movement of said intermediate sequence of bits into a destination register or a source register and two configuration registers are used for storing said configuration bits.

55. A system for performing a permutation of a source sequence of bits in a programmable processor comprising:

means for defining an intermediate sequence of bits that said source sequence of bits is transformed into using an interconnection network, and means for determining a sequence of permutation operations for transforming said source sequence of bits into one or more intermediate sequence of bits until a desired sequence of bits is obtained.

56. A system for performing a permutation of a source sequence of bits in a programmable processor comprising:

means for defining an intermediate sequence of bits that said source sequence of bits is transformed into using omega and flip network stages; and

means for determining a sequence of permutation operations for transforming said source sequence of bits into one or more intermediate sequence of bits until a desired sequence of bits is obtained.

57. The system of claim 56 wherein said intermediate sequence of bits is determined with a configuration of an omega-flip network, said omega-flip network comprising at least two stages wherein each said stage is either an omega network stage or a flip network stage.

58. The system of claim 56 wherein at most $2\lg r/m$ said permutation instructions are included in said permutation instruction sequence, wherein r is the number of subwords in said sequence of bits, each said subword comprising one or more bits, and m is the number of network stages performed by one permutation instruction.

59. A computer implemented method for performing an arbitrary permutation of a sequence of bits comprising the steps of:

inputting said sequence of bits into a source register;

connecting said source register to an omega-flip network;

in response to an omega-flip instruction selecting a configuration of said omega-flip network; and

moving each of said bits in said source register to a position in a sequence of bits of a destination register based on configuration bits of a configuration register.

60. The method of claim 59 wherein at most 2lg*r*/*m* said permutation instructions are included in said permutation instruction sequence, wherein *r* is the number of subwords in said sequence of subwords in said sequence of bits, each said subword comprising one or more bits, and *m* is the number of network stages performed by one permutation instruction.

61. A computer system for performing an arbitrary permutation comprising:

a source register;

a configuration register;

a destination register;

an omega-flip network coupled to said first source register, said configuration register and said destination register, in response to an omega-flip instruction selecting a configuration of said omega-flip network, placing each bit in said sequence of bits from said source register to a position in a sequence of bits in said destination register based on a configuration of bits of said configuration register.

62 The system of claim 61 wherein at most 2lg*r*/*m* said permutation instructions are included in said permutation instruction sequence, wherein *r* is the number of subwords in said sequence of subwords in said sequence of bits, each said subword comprising one or more bits, and *m* is the number of network stages performed by one permutation instruction.

63. A computer readable medium having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more intermediate sequences of subwords, each intermediate sequence of subwords using input from said source sequence of subwords or a previous said intermediate sequence of subwords until a desired sequence of subwords is obtained, wherein each subword is one or more bits, said intermediate sequence being determined using a multistage interconnection network.

64. A computer readable medium having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more

intermediate sequences of subwords, each intermediate sequence of subwords using input from said source sequence of subwords or a previous said intermediate sequence of subwords until a desired sequence of subwords is obtained, wherein each subword is one or more bits, said intermediate sequence being determined using an omega and flip network stages.

65. The computer readable medium of claim 64 wherein said omega-flip network comprises at least two stages wherein each stage is either an omega network stage or a flip network stage.

66. The computer readable medium of claim 64 wherein at most 2lg*r*/*m* said permutation instructions are included in said permutation instruction sequence, wherein *r* is the number of subwords in said sequence of subwords in said sequence of bits, each said subword comprising one or more bits, and *m* is the number of network stages performed by one permutation instruction.

09850238.050701

67. A cryptographic system, having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more intermediate sequences of subwords, each intermediate sequence of subwords using input from said source sequence of subwords or a previous said intermediate sequence of subwords until a desired sequence of subwords is obtained, wherein each subword is one or more bits, said intermediate sequence being determined using a multistage interconnection network.

68 A cryptographic system, having stored thereon data representing a sequence of permutation instructions, the sequence of permutation instructions which when executed by a processor, cause the processor to permute a source sequence of subwords into one or more intermediate sequences of subwords, each intermediate sequence of subwords using input from said source sequence of subwords or a previous said intermediate sequence of subwords until a desired sequence of subwords is obtained, wherein each subword is one or more bits, said intermediate sequence being determined using an omega and flip network stages.

69. The cryptographic system of claim 68 wherein the omega-flip network comprises at least two stages wherein each stage is either an omega network stage or a flip network stage.

70. The cryptographic system of claim 68 wherein at most 2lg*r*/*m* said permutation instructions are included in said permutation instruction sequence, wherein *r* is the number of subwords in said sequence of subwords in said sequence of bits, each said subword comprising one or more bits, and *m* is the number of network stages performed by one permutation instruction.

09850238.050201